# Introduction to Software Engineering

CSCI 577a

LCA Workshop

# The Road Ahead: LCA

Success criteria:

**Commit to constructing a feasible architecture**

# Pre-wedding plans

- What do you need to do between LCO and LCA?
  - Design
    - Resolve all significant architectural issues
    - Advanced prototyping
  - Refine
    - Resolve model clashes
    - Add significant details
    - Remove non-significant details - no “fluff”
    - Schedule, roles, commitments, effort estimates, etc.
  - Justi-fine! (Justify)
    - Assure architecture is faithful to concept
    - Assure value of system vs. stakeholder investment
    - Reduce risk exposure

# What does this involve?

- Several iterations through MBASE models
  - Model integration and integrity paramount
  - Some CTS stuff **now**
- Refine WinWin negotiations
  - Close out old issues
  - Cover all win conditions
  - Identify and deal with new win conditions
- Writing code
  - Advanced prototyping to resolve risky architectural issues
  - Head start on implementing critical requirements (for assurance, schedule, etc.)

# Life Cycle Architecture (LCA)

- More formal, with everything appropriate specifically tracing upward and downward
- No major unresolved issues or items, and closure mechanisms identified for any unresolved issues or items (e.g., “detailed data entry capabilities will be specified once the Library chooses a Forms Management package on February 15”)
- No more TBD's
- There should no longer be any "possible" or "potential" elements (e.g., Entities, Components, …)
- No more superfluous, unreferenced items: each element (e.g., Entities, Components, …) either should reference, or be referenced by another element. Items that are not referenced should be eliminated, or documented as irrelevant

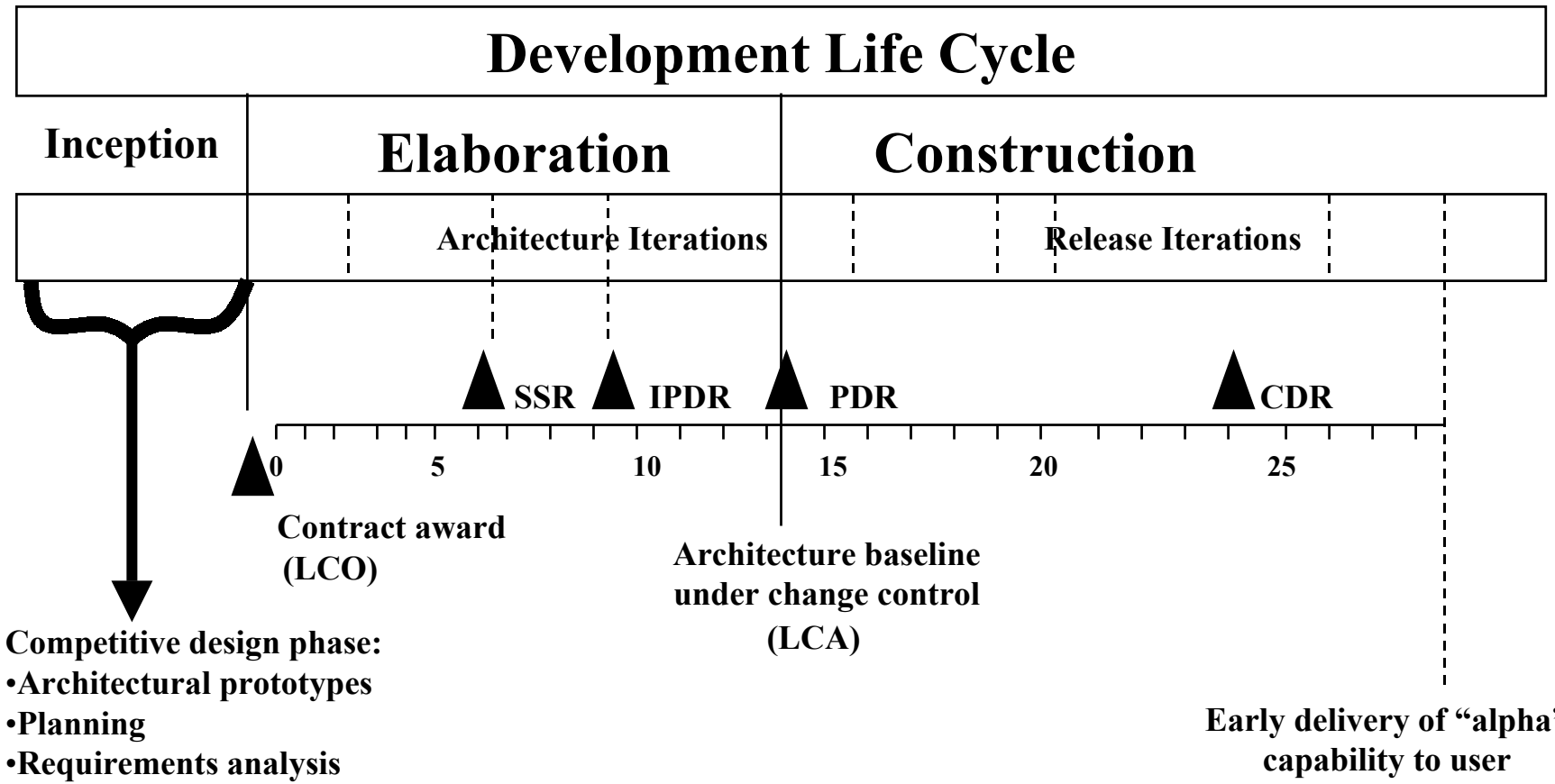
# MBASE Milestone Elements

| Milestone Element | Life Cycle Objectives (LCO) | Life Cycle Architecture (LCA) |
|---|---|---|
| **Definition of Operational Concept** | • Top-level system objectives and scope<br>- Shared vision of expected initiatives and outcomes [COTS strategy for reuse and legacy elements]<br>- System boundary [major COTS component boundaries within system]<br>- Environment parameters and assumptions [CBS success factors]<br>- Evolution parameters [major COTS vendor product availability, upgrade cycles]<br>• Operational concept<br>- Operations and maintenance scenarios and parameters [COTS maintenance]<br>- Organizational life-cycle responsibilities (stakeholders) [major COTS vendors] | • Elaboration of system objectives and scope by increment<br>• Elaboration of operational concept by increment [COTS operation and use summary, vendor support strategies, COTS adoption impact on organization] |
| **System Prototype(s)** | • Exercise key usage scenarios [demonstrate key COTS capabilities and validate interfaces, scope use and value of major COTS products]<br>• Resolve critical risks [COTS selection and evaluation plan] | • Exercise range of usage scenarios [complete COTS evaluations, initial COTS configuration and training, initial COTS tailoring]<br>• Resolve major outstanding risks [resolve COTS integration issues and level of use] |
| **Definition of System Requirements** | • Top-level functions, interfaces, quality attribute levels, including:<br>- Growth vectors<br>- Priorities<br>- Legacy systems and environments<br>- [Establish key COTS evaluation attributes and screening parameters]<br>• Stakeholders' concurrence on essentials [mandated COTS packages, suppliers and vendors, identify negotiable and non-negotiable COTS requirements, COTS evaluation win-conditions and constraints] | • Elaboration of functions, interfaces, quality attributes by increment<br>- Identification of TBDs (to-be-determined items) [mapping of COTS capabilities to requirements]<br>• Stakeholders' concurrence on their priority concerns [concurrence on COTS imposed requirements and 90% re-worked requirements tradeoffs] |

# MBASE Milestone Elements (2)

| Milestone Element | Life Cycle Objectives (LCO) | Life Cycle Architecture (LCA) |
|---|---|---|
| **Definition of System and Software Architecture** | • Top-level definition of at least one feasible architecture<br>- Physical and logical elements and relationships, including top-level domain factoring and identification of possible design patterns<br>- Choices of COTS and reusable software elements [mapping of critical system components to COTS products]<br>• Identification of infeasible architecture options | • Choice of architecture and elaboration by increment<br>- Physical and logical components, connectors, configurations, constraints<br>- COTS, reuse choices [CBS design (mapping of components and system factors to COTS), COTS package configurations]<br>- Domain-architecture and architectural style choices [design for COTS glue code and wrappers]<br>• Architecture evolution parameters |
| **Definition of Life-Cycle Plan** | • Identification of life-cycle stakeholders<br>- Users, customers, developers, maintainers, interpreters, general public, others [COTS suppliers and vendors]<br>• Identification of life-cycle process model<br>- Top-level stages, increments [COTS deployment – prototyping, evaluation, schedule, risks impact of schedule and cost, stakeholders, license negotiations]<br>• Top-level WWWWWHH* by stage [COTS product lifecycles (releases, delivery dates, costs, training needs) | • Elaboration of WWWWWHH* for Initial Operational Capability (IOC)<br>- Partial elaboration, identification of key TBDs for later increments<br>- [COTS upgrades, patches]<br>- [License managament (non-strandard provisions)]<br>- [COTS users, maintainer, and developer skill set attributes]<br>- [COTS training plan]<br>- [COTS risks realized; integration, cultural, economic] |
| **Feasibility Rationale** | • Risk assessment and mitigation plans [COTS vendor issues – product availability, stability, costs]<br>• Assurance of consistency among elements above<br>- Via analysis, measurement, prototyping, simulation, etc.<br>- Business case analysis for requirements, feasible architectures [buy versus build rationale, justification of CBS strategy, CBS tradeoffs] | • Assurance of consistency among elements above [feasibility of COTS vendor relationships, COTS maintenance costs, organization readiness to implement COTS]<br>• All major risks resolved or covered by risk management plan [completed license contracts, COTS product alternatives and marketplace issues, COTS organization culture issues, vendor commitments] |

# Common Subsystem Macroprocess



# The rest of the project….

- Remember the four MBASE project phases:
  - Inception
  - Elaboration
  - Construction
  - Transition
  - (Then you spend your life supporting it.)
- Elaboration ends approximately at the time of your LCA ARB. Then, construction begins after RLCA in CS577b (though you may start sooner than that!).

# Get ready for the LCA ARB…

Tasks for LCA:

1. Finalize your OCD, SSRD.
2. Get your SSAD to a point where you can construct something from it.
3. Prove to us that you're organized enough to finish on time, in your LCP.
4. Use your FRD to prove that the rest of the documents are sane, and that your project will work.

*Remember that an ARB is just a checkpoint and feedback review so do not hide the "dirty laundry"*

# LCA ARB Session Overview

- **Less time for OCD, Prototype**
- **More time for Architecture, Plan**
- **Details TBD based on LCO ARB experience**
- **Focus on changes since LCO**
- **Emphasize material that is relevant to 577B (or to end of class)**
- **Risk management still fundamental**

# LCA ARB Session Outline

**(x,y): (presentation time, total time)**
**(10,15) OCD. Prototype update**
**(5,10) Requirements. Most significant requirements**
**(15,20) Architecture. Overall design, COTS/reuse choices**
**(15,20) Life Cycle plan for 577b or as appropriate to project**
**(5,10) Feasibility Rationale. Refined business case; major risks; Requirements satisfaction, process rationale and consistency**
**(0,5) Things done right; issues to address (Instructor)**

- **Plan on 2 minutes per briefing chart, except title**
- **Focus on changes (particularly new things) since LCO**
- **You may vary from the above, but please plan and notify ARB board members in advance**

# ARB Chartsmanship

- Don’t repeat the MBASE Guidelines
- Don’t sweat the small stuff
- Use audience-based terminology
- Assume 2 minutes presentation time per chart
  - After timed dry run practice
- Don’t repeat previous speakers’ material
  - OK to refer to it
- Do dry runs with outsider audience

# LCA 1: Finalize OCD, SSRD

- Finalize your OCD, SSRD so that your other documents can build on a solid base.
  - Now that you've had one review, work out remaining open requirements with your customer. This may not be completely possible, so do your best.
  - Remember that your OCD/SSRD can and should be agnostic toward specific implementations: get the "what" of your project down, leave the "how" to the SSAD.
  - Probably your OCD and SSRD won't have to change much between LCO and LCA. But if they do need to, make the changes soon, and show them to the rest of your group!

# LCA 2: Finish your SSAD, 1

- Your SSAD is the most critical document in your LCA package/presentation.
  - Remember, your goal is to demonstrate to use ONE architecture that your group can build and will satisfy the requirements.
  - You may have had architecture options at LCO. By LCA, decide on one. You make this decision based on your customer's input, advice from your TAs, and (in particular) a series of risk-driven prototypes (risk-driven means your prototype the hard parts).

# LCA 2: Finish your SSAD, 2

- How to decide if your SSAD is good?
  - Your SSAD is doing the right thing when it accurately maps the SSRD onto a set of components, each with a specific design.
  - Your teams programmers should be able to take the SSAD and implement each object, making the ‘how’ decisions in program code without wasting thought on the ‘what’ of each object, and with confidence that the objects will, when assembled together, yield the finished product. (That is, your architecture should keep your programmers from thinking too much.)
  - A good SSAD directly inspires your construction schedule: dependencies should be clear, so you can identify which pieces have to be built first, and which can be built in parallel.

# LCA 3: LCP, 1

- Your LCP is critical to finishing your product once you know how to do it (SSAD).
- Identify a few (2-5) *iterations*, or phases:
  - For each, identify specific tasks for each member, by name and role ("In iteration 2, Fred will be constructing the User Interface API while Jane tests the DB interface constructed in iteration 1 (see SSAD 3.x.x. and IOC Test Plan 2.x, 2.y).")
  - For each, identify milestones (finished objects and components, test results, reports, reviews, deliverables, meetings).
  - Set durations, and tentative specific dates, as best you can.

# LCA 3: LCP, 2

- Things for specific sections:
  - In 4.1, “Risk Management”, identify the schedule effects (in terms of durations and dates of your iterations) if a risk item happens.
  - In 4.3, “Reviews”, you’re being prompted to schedule reviews other than the mandated LCO, LCA, IOC. Schedule formal or informal reviews of specific aspects of your project. Reviews might include your customer or a TA, or might be internal reviews among a subset of your group. The ends of iterations are good times for internal reviews.
- In general, be as specific, but realistic, as you can. This hard, but very important (and it will make your life easier in the long run….)

# LCA 4: FRD

- Use the FRD to validate your process. In it, you can show how cool you are. And, you can cover your ---.
  - In the business case analysis (2.1), don't think that your project costs nothing just because the customer doesn't have to pay out any money for it. It costs your time, and some of the customers. Show that the time expended will be worthwhile.
  - In the "requirements satisfaction" section (2.2), show specifically how your system will implement the requirements, by referencing (in a table?) the SSRD and SSAD.
  - In section 3, "process rationale", show that your schedule and organization and your contingency plans are in line with the requirements…and the really core requirements.

# LCA "Checklist"

*** Warning ***

This checklist serves as a checkpoint to remind you of a few qualities each MBASE deliverable in your LCA package must have. It is NOT an exhaustive list and satisfaction of all the items listed does NOT indicate that your LCA package is complete and satisfies all the LCA milestone requirements.

# OCD LCA Checks

1. Organization Goals are all clearly documented as M.R.
2. Project Goals are M.R.S. with respect to the Organization Goals and Organization Activities
3. Quality Goals are all clearly documented as M.R.S. with respect to System Responsibilities
4. All Organization Goals are referenced by something later such as Project Goals, System Responsibilities (or they should be excluded are documented as irrelevant)
5. All Organization Activities are referenced by something later such as System Responsibilities, Project and Quality Goals, however just about anything can reference (or they should be excluded are documented as irrelevant)
6. All Entities have specification templates filled out (possible Entities are not listed in LCA deliverables)
7. All Entities are referenced by something (usually Components or other Entities for which there are dependencies)
8. Project Goals, Quality Goals reference and are referenced by other model elements (Project Goals reference Organization Goals or Activities and are referenced by Project Requirements. Quality Goals usually reference System Responsibilities and sometimes Organization Goals or Activities and are referenced by Quality Attribute Requirements.)
9. All System Responsibilities reference and are referenced by other elements (Behaviors, Requirements, and Quality goals usually)
10. System Responsibilities are not behaviors or operations and are exactly the same as those used in the SSAD Behavior model.
11. Statement of Purpose indicates relation to Organization Background
12. CDL for uncommon undefined terms in the Domain Description
13. Operational Scenarios have specifications and reference System Responsibilities and goals.

# SSRD LCA Checks

1. System Definition refers to Statement of Purpose
2. A clear block diagram consistent with the SSAD design views
3. Project Requirements reference Project Goals and are M.A.R.S have specifications that are consistent with SSAD design
4. Quality Attribute Requirements reference Quality Goals and System Requirements and are M.A.R.S (and Requirements Satisfaction in FRD)
5. System Requirements reference System Responsibilities, Win Conditions, Project Goals, Quality Goals, and Project Requirements (and are referenced by Operations in SSAD and Requirements Satisfaction in FRD)
6. Both nominal and off-nominal requirements have clear implementation specifications
7. Each System Requirement can be implemented in a consistent manner with respect to the SSAD design has a testable Use-Case scenario
8. Interface Requirements have clear specifications based on OCD Scenarios.
9. All requirements are defined in such as way that they can be implemented tested (this includes Evolutionary Requirements and Interface Requirements)
10. All requirements have been prioritized and challenged as to their necessity

# SSRD LCA Checks

1. System Definition refers to Statement of Purpose
2. A clear block diagram consistent with the SSAD design views
3. Project Requirements reference Project Goals and are M.A.R.S have specifications that are consistent with SSAD design
4. Quality Attribute Requirements reference Quality Goals and System Requirements and are M.A.R.S (and Requirements Satisfaction in FRD)
5. System Requirements reference System Responsibilities, Win Conditions, Project Goals, Quality Goals, and Project Requirements (and are referenced by Operations in SSAD and Requirements Satisfaction in FRD)
6. Both nominal and off-nominal requirements have clear implementation specifications
7. Each System Requirement can be implemented in a consistent manner with respect to the SSAD design has a testable Use-Case scenario
8. Interface Requirements have clear specifications based on OCD Scenarios.
9. All requirements are defined in such as way that they can be implemented tested (this includes Evolutionary Requirements and Interface Requirements)
10. All requirements have been prioritized and challenged as to their necessity

# SSAD LCA Checks

1. Components reference Entities
2. Objects reference Components
3. Each Component has a design in Objects (or is detailed as a Design-Component-Object such as COTS packages)
4. Components and Objects have specification templates filled out (no possible Components or Objects in LCA)
5. Behaviors start with System Responsibilities from OCD (an exact match, if they're not workable for behaviors then the architect should reconcile them with the OCD person), and reference Project Goals, Quality Goals where necessary
6. Each Operation references Behaviors and elaborate System Requirements and Quality Attribute Requirements
7. At least one Enterprise Class in the Enterprise Model for each Component
8. All components mapped into Logical Design View, System Layer View, and Deployment View
9. At least one Object Class in the Class Model for each Object (Class taxonomies separated into implementation types)
10. All Operations in Operation Model (Rose sequence diagrams) are assigned to Objects that are detailed in the Object Model
11. All Operations across objects have either an outlet relation specified in the object model or a dynamically created reference

# LCP LCA Checks

1. Milestones and schedules covering all core development areas (especially 577b areas)
2. Process model elaborations per phase (what is done on Inception, Elaboration, Construction, transition)
3. Top-n risks and impacts on schedule and budget
4. Staff assignments to work breakdown structure
5. Detailed project deliverables covering all core development areas
6. Demonstrated quality assurance procedures
7. Demonstrated configuration management procedures
8. Detailed cost and effort estimates

# FRD LCA Checks

1. Fleshed out business case (no TBD's) – all major costs accounted for and converted to comparable units (time, dollars, utilities, reputation, etc. as appropriate)
2. Value added relates to OCD project goals, organization goals, benefits realized, and results chain outcomes
3. Comparison of savings of ongoing or expected proposed system costs versus current system
4. Elaboration of return on investment (value minus initial and ongoing costs) over expected lifetime of system
5. Tracing of SSRD requirements to OCD capabilities and goals with rationale as to how they are related (usually a very straightforward mapping) to indicate operational concept satisfaction
6. Tracing of SSAD design elements to SSRD requirements with rationale as to how they are related to indicate requirements satisfaction
7. System priority sets with references to SSRD requirements or project development activities as appropriate that cover core development areas
8. Elaboration of how process choices in LCP are compatible with system priorities
9. Elaboration of how process choices will handle system priorities given the resources detailed in LCP
10. All outstanding risks referenced from LCP have risk assessments, mitigation procedures, and contingency plans with impacts

# CTS LCA Checks

1. Initial Increment Plan implementing LCP schedules and milestones compatible with development process model choice
2. Initial Test Plan covering critical SSRD requirement areas
3. Initial Quality Management Plan implementing LCP quality assurance activities
4. Inspection Plan and Inspection Reports as indicated in LCP

# Model Integrity LCA Checks

1. All risky items elaborated and explicitly risk managed
2. Removal of superfluous elements and unnecessary detail (as appropriate to model audience)
3. Specific references of all directly related elements within and between models (e.g. reference particular OCD Project Goals for particular SSRD Project Goals)
4. Consistency of documentation look and feel

Conformance to MBASE guidelines or rationale for departures or variations

# The Road Ahead Ahead: IOC

Success criteria

**The initial operational capabilities of the system as constructed satisfy the architecture models**

* This includes all the MBASE models, not just SSAD

** Completeness is not as important as soundness